Министерство образования и науки Российской Федерации

(МИНОБРНАУКИ РОССИИ)

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)

Институт прикладной математики и компьютерных наук

Кафедра защиты информации и криптографии

**КУРСОВАЯ РАБОТА**

БИБЛИОТЕКА ДЛЯ РАБОТЫ С БУЛЕВЫМИ ФУНКЦИЯМИ ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ LYAPAS

Муругов Михаил Алексеевич

Руководитель

канд. физ.-мат. наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_И.А.Панкратова «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_201\_\_\_г.

Студент группы № 1155

\_\_\_\_\_\_\_\_\_\_\_\_\_М.А.Муругов

Томск 2018

ОГЛАВЛЕНИЕ

Введение………………………………………………………………………………………….2

1. Описание алгоритмов на математическом языке…………………………………………..3
   1. Принадлежность булевой функции к классу 
   2. Принадлежность булевой функции к классу 
   3. Принадлежность булевой функции к классу монотонных булевых функций
   4. Преобразование Мёбиуса булевой функции
   5. Принадлежность булевой функции к классу линейных булевых функций
   6. Отражение вектора значений булевой функции
   7. Принадлежность булевой функции к классу самодвойственных булевых функций
2. Идеи программных реализаций
   1. Принадлежность булевой функции к классу 
   2. Принадлежность булевой функции к классу 
   3. Принадлежность булевой функции к классу монотонных булевых функций
   4. Преобразование Мёбиуса булевой функции
   5. Принадлежность булевой функции к классу линейных булевых функций
   6. Отражение вектора значений булевой функции
   7. Принадлежность булевой функции к классу самодвойственных булевых функций
3. Экспериментальные данные
   1. Принадлежность булевой функции к классу 
   2. Принадлежность булевой функции к классу 
   3. Принадлежность булевой функции к классу монотонных булевых функций
   4. Преобразование Мёбиуса булевой функции
   5. Принадлежность булевой функции к классу линейных булевых функций
   6. Отражение вектора значений булевой функции
   7. Принадлежность булевой функции к классу самодвойственных булевых функций
4. Заключение

Список использованных источников и литературы

Приложения

**ВВЕДЕНИЕ**

Целью этой курсовой работы было написание библиотеки для работы с булевыми функциями (определение принадлежности к замкнутым классам, различные преобразования и т.д.) для языка программирования LYaPAS. В дальнейшем планируется, что эта библиотека будет использоваться для реализации криптографических алгоритмов и прочих нужд.

Для криптографии булевы функции важны т.к. они, в частности, используются в качестве комбинирующих и фильтрующих функций при построении поточных шифров; для блочных шифров они используются в качестве функций блоков замены и т.д.

В нынешнее время язык LYaPAS уже выигрывает по скорости на некоторых алгоритмах, но всё ещё требует доработок и улучшений, в следствие чего и был выбран.

В качестве базиса в языке уже реализованы побитовые операции для булевых векторов любой длины, а также для 32-х битных векторов функция подсчёта веса и генерация псевдослучайного вектора. Всё это используется в настоящей работе для реализации более сложных вещей относительно булевых функций.

**ОПИСАНИЕ АЛГОРИТМОВ НА МАТЕМАТИЧЕСКОМ ЯЗЫКЕ**

***Принадлежность булевой функции к классу ***

**Определение.** Булева функция *сохраняет константу 0* (*принадлежит классу* *)*, если на наборе из всех нулей функция принимает значение нуль.

Алгоритм:

Вход:  – булева функция

Выход: “ принадлежит классу ?”

Шаг 1) Если , то ответ “Да”

Иначе ответ “Нет”

***Принадлежность булевой функции к классу ***

**Определение.** Булева функция *сохраняет константу 1* (*принадлежит классу* ), если на наборе из всех единиц функция принимает значение единица.

Алгоритм:

Вход:  – булева функция

Выход: “ принадлежит классу ?”

Шаг 1) Если , то ответ “Да”

Иначе ответ “Нет”

***Принадлежность булевой функции к классу монотонных булевых функций***

**Определение.** Булева функция  называется *монотонной* (*принадлежит классу* ), если для любой пары наборов  и  таких, что , выполняется условие .

Алгоритм определения принадлежности булевой функции к классу монотонных булевых функций приведён в [1].

***Преобразование Мёбиуса булевой функции***

**Определение.** *Положительной конъюнкцией* называется элементарная конъюнкция, не содержащая инверсий переменных. Договоримся обозначать положительную конъюнкцию через .

**///Определение АНФ взять из “Булевы функции в криптографии”! (не нашёл)**

**Определение.** *Полиномом Жегалкина*, или *алгебраической нормальной формой (АНФ)*, булевой функции  называется дизъюнкция с исключением различных положительных конъюнкций переменных из множества , то есть формула вида , задающая функцию .

**Определение.** *Преобразованием Мёбиуса* называется функция , где  – множество всех булевых функций от переменных. С помощью преобразования Мёбиуса решается задача построения АНФ булевой функции, и вычислить его значения для функции  можно по формуле , где .

Преобразование Мёбиуса связано с полиномом Жегалкина следующим образом: значение  на наборе аргументов говорит о том, есть ли положительная конъюнкция аргументов со значением  из этого набора в АНФ функции  ( – положительная конъюнкция есть,  – положительной конъюнкции нет). Набор аргументов  соответствует константе .

Алгоритм:

Вход:  – булева функция

Выход: 

Шаг 1) 

Шаг 2) Для всех :

Шаг 2.1) 

***Принадлежность булевой функции к классу линейных булевых функций***

**Определение.** *Длиной* булева вектора назовем количество его компонент, а *весом* вектора – количество компонент, равных единице

Длину булева вектора  в дальнейшем будем обозначать . Запись , где  – булева функция, будет обозначать длину вектора её значений.

Вес булева вектора  в дальнейшем будем обозначать . Запись , где  – булева функция, будет обозначать вес вектора её значений.

**Определение.** *Длиной полинома Жегалкина* назовем количество конъюнкций в полиноме, а его *степенью* – наибольший из рангов конъюнкций, входящих в полином.

**Определение.** Полином Жегалкина называется *линейным*, если его степень не превышает единицы.

**Определение.** Булева функция называется *линейной* (*принадлежит классу* ), если ее полином Жегалкина линеен.

Алгоритм:

Вход:  – булева функция

Выход: “ – линейна?”

Шаг 1) 

Шаг 2) Если полином Жегалкина, построенный по коэффициентам  линеен, то ответ “Да”

Иначе ответ “Нет”

***Отражение вектора значений булевой функции***

**Определение.** *Отражением* вектора значений булевой функции является обмен значениями на противоположных наборах аргументов.

В дальнейшем отражение вектора значений булевой функции  будем обозначать 

Алгоритм:

Вход:  – булева функция

Выход: 

Шаг 1) Для всех  таких, что :

Шаг 1.1) 

***Принадлежность булевой функции к классу самодвойственных булевых функций***

**Определение.** Булева функция  называется *двойственной булевой функции* , если она получена из  инверсией всех аргументов и самой функции, то есть .

**Определение.** Булева функция  *самодвойственна* (*принадлежит классу* ), если она равна двойственной себе функции, то есть .

Алгоритм:

Вход:  – булева функция

Выход: “ – самодвойственна?”

Шаг 1) Для всех векторов  таких, что :

Шаг 1.1) Если , то ответ “Нет”

Шаг 2) Ответ “Да”

**ИДЕИ ПРОГРАММНЫХ РЕАЛИЗАЦИЙ**

Перед изложением дальнейшего материала необходимо кое-что обозначить:

Во-первых, булевы функции в языке LYaPAS представляются векторами их значений.

Во-вторых, вектора значений булевых функций хранятся в логических комплексах L, каждый элемент которого занимает в памяти 4 байта(32 бита). Таким образом, т.к. , то функция до 5 аргументов включительно помещается в один элемент комплекса. От 6 в 2 элемента, от 7 в 4 и т.д. Количество элементов комплекса, необходимых для хранения функции от  аргументов можно вычислить по формуле .

В-третьих, значения булевой функции в памяти хранятся в привычном нам порядке (младшие биты справа, старшие биты слева), при этом нулевой бит нулевого элемента комплекса соответствует значению , следующий за ним  и т.д.

Также обратите внимание, что в этом разделе находятся лишь идеи, сами программные реализации смотрите в приложении.

***Принадлежность к классу ***

Проверка булевой функции на принадлежность к классу  тривиальна. Необходимо просто посмотреть на первый бит вектора её значений. Если этот бит равен нулю, то функция сохраняет константу 0.

***Принадлежность к классу ***

Для проверки принадлежности булевой функции к классу  необходимо посмотреть на старший бит вектора её значений. Если этот бит равен 1, то функция сохраняет константу 1. Но проверка булевой функции на принадлежность к классу  немного сложнее, чем к классу , т.к. у функций, зависящих от  аргументов старший бит вектора значений находится в нулевом элементе комплекса и его сначала необходимо найти. В общем же случае найти старший бит вектора значений функции можно по следующим правилам: , , где  – индекс элемента комплекса, а  – номер бита в элементе с индексом .

***Принадлежность булевой функции к классу монотонных булевых функций***

Т.к. булева функция помещена в «блоки» по 32 бита, то перед стартом рекурсии можно проверить её на монотонность вплоть до 5 компоненты следующими действиями:

L1i < 1 & AAAAAAAAh ⇒ a \*\*\*Проверяем на монотонность на наборах,

a & L1i ⊕ a ↦2 \*\*\*соседних по пятой компоненте

L1i < 2 & CCCCCCCCh ⇒ a \*\*\*Проверяем на монотонность на наборах,

a & L1i ⊕ a ↦2 \*\*\*соседних по четвёртой компоненте

L1i < 4 & F0F0F0F0h ⇒ a

a & L1i ⊕ a ↦2 \*\*\*...

L1i < 8 & FF00FF00h ⇒ a

a & L1i ⊕ a ↦2

L1i < 16 ⇒ a \*\*\*Проверяем на монотонность на наборах,

a & L1i ⊕ a ↦2 \*\*\*соседних по первой компоненте

После того, как каждый элемент комплекса проверен таким образом и немонотонность не обнаружена, то запускается рекурсивная функция, которая проверяет на монотонность по следующим компонентам.

***Преобразование Мёбиуса булевой функции***

Как следует из способа, изложенного в [2], преобразование Мёбиуса рекурсивно реализуется по следующему алгоритму:

Шаг 1) Разбиваем вектор значений булевой функции на младшую и старшую часть  и  соответственно

Шаг 2) 

Шаг 3) Если , то выход

Иначе выполнить эту процедуру для  и 

***Принадлежность булевой функции к классу линейных булевых функций***

Алгоритм на проверку принадлежности булевой функции довольно прост. Необходимо выполнить преобразование Мёбиуса для этой функции и посмотреть, есть ли хотя бы одна единица на наборе аргументов с более, чем одной единицей.

Алгоритм:

Вход:  – булева функция

Выход: “ – линейна?”

Шаг 1) 

Шаг 2) 

Шаг 3) Для всех  таких, что :

Шаг 3.1) 

Шаг 4) Если , то ответ “Да”

Иначе ответ “Нет”

***Отражение вектора значений булевой функции***

Отражение вектора значений булевой функции программно довольно нетривиально, т.к. нет таких средств, которые позволили бы сделать это за одну операцию. В языке LYaPAS, т.к. вектора значений булевых функций разбиты на «блоки» по 32 бита, преобразование выполняется следующим образом: сначала выполняется отражение каждого «блока» по отдельности, затем первый «блок» меняется местами с последним, второй с предпоследним и т.д.

Также, т.к. вектора значений булевых функций от  переменных включительно помещаются в один элемент логического комплекса L, то после отражения таких векторов их необходимо будет побитово сдвинуть вправо на  бита, т.к. после отражения младшие биты станут старшими в 32-х битном «блоке».

Функция, выполняющая отражение 32-х битного «блока»:

reverseBits(a/b)

\*\*\*a – входной вектор

\*\*\*b – отраженный вектор a

a > 1 & 55555555h ⇒ v

a & 55555555h < 1 ∨ v ⇒ b \*\*\*Меняем местами чётные и нечётные биты

b > 2 & 33333333h ⇒ v

b & 33333333h < 2 ∨ v ⇒ b \*\*\*Меняем местами пары битов

b > 4 & 0f0f0f0fh ⇒ v

b & 0f0f0f0fh < 4 ∨ v ⇒ b \*\*\*Меняем местами последовательности из 4-х битов

b > 8 & 00ff00ffh ⇒ v

b & 00ff00ffh < 8 ∨ v ⇒ b \*\*\*Меняем местами байты

b > 16 ⇒ v

b < 16 ∨ v ⇒ b \*\*\*Меняем местами 2-х байтовые слова

\*\*

***Принадлежность булевой функции к классу самодвойственных булевых функций***

Программно проверка на принадлежность булевой функции к классу самодвойственных булевых функций реализована согласно следующему алгоритму:

Вход:  – булева функция

Выход: “ – самодвойственна?”

Шаг 1) Разбиваем вектор значений булевой функции на младшую и старшую часть  и  соответственно

Шаг 2) Если , то ответ “Да”

Иначе ответ “Нет”

 – этим самым действием мы сопоставляем значения функции на противоположных наборах аргументов, а затем проверяем условие .

**ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ**

Ниже будут представлены графики зависимости времени работы функций над булевыми функциями от количества переменных в булевых функциях. Для каждого количества переменных выполнялось по 100000 итераций и на вход подавался наихудший случай (проверка констант на монотонность и т.д.).

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ**

1. Быкова С.В. Учебно-методический комплекс «Булевы функции». Томск 2006.
2. Панкратова И.А. Учебное пособие «Булевы функции в криптографии». Томск 2014.